Reti di Calcolatori

Esercitazioni di laboratorio

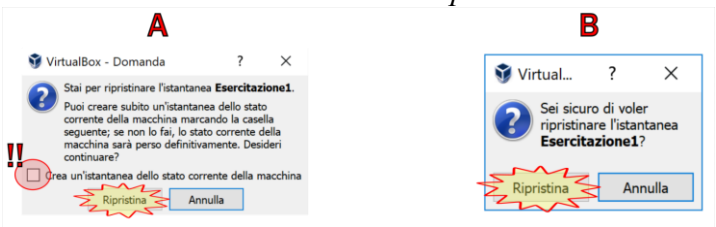
Le seguenti esercitazioni hanno lo scopo di far prendere dimestichezza con una struttura di rete elementare, configurando funzioni e servizi fondamentali che costituiscono l'ossatura delle moderne reti di calcolatori, ovvero routing table, DHCP, DNS, server web e mail. Contestualmente verrà proposta un'analisi del traffico di rete in modo da osservare nel dettaglio il funzionamento dei protocolli coinvolti.
Le varie esercitazioni sono "indipendenti" tra loro ma si consiglia lo svolgimento secondo sequenza, la configurazione iniziale di una esercitazione corrisponde alla conclusiva di quella precedente.

Informazioni di base

Le esercitazioni si svolgono su un'architettura virtuale appositamente configurata con VirtualBox. Ogni esercitazione fa uso di una propria configurazione hardware e software memorizzata in opportuni snapshot dei sistemi guest[1] e **non va assolutamente modificata**, pena il corretto svolgimento dell'esercitazione.
Per procedere è necessario attivare per tutti i sistemi guest lo snapshot relativo all'esercitazione desiderata.

1. Selezionare la VM (ad es. *R1*) e la sezione *istantanee/snapshot.*
2. Selezionare lo snapshot per l'esercitazione desiderata (ad es. *Esercitazione1*)
3. Click su icona *Ripristina istantanea/snapshot*

4. In base allo stato corrente del sistema guest selezionato possono venire visualizzate due finestre di dialogo differenti. Nel caso "*A*" **deselezionare** l'opzione "*Crea istantanea dello stato corrente della macchina*" e click su *Ripristina*.



[1]Nell'ambito della virtualizzazione si denota con 'host' il sistema fisico e con 'guest' il sistema virtualizzato; il termine host verrà comunque anche usato per indicare un generico sistema sia esso fisico o virtuale.

5. Ripetere i punti precedenti per le altre VM in modo che tutte facciano riferimento alla configurazione della medesima esercitazione come l'esempio in figura.

6. Avviare le VM. Per l'avvio è possibile effettuare click con il tasto destro sul nome del gruppo "*Reti_di_Calcolatori-ESERCITAZIONE*" e selezionare *"Avvia → Avvio normale"*

E' possibile svolgere tutta l'esercitazione con il proprio PC installando VirtualBox© e scaricando dal sito Ladispe sezione "*Corsi -> Reti di Calcolatori*" il file contenente le VM preconfigurate. Durante l'importazione, se richiesto, fare attenzione a **NON inizializzare i MAC** address delle varie schede di rete.

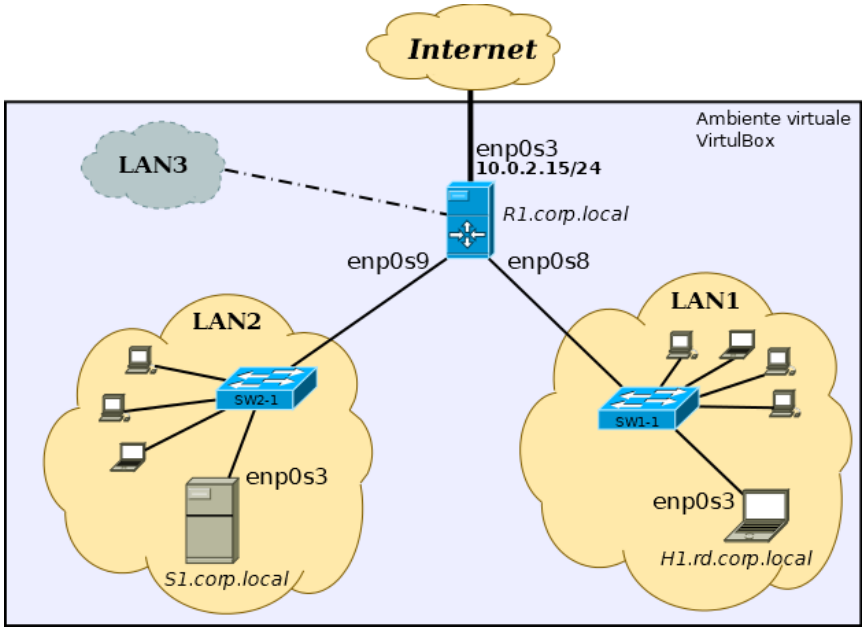
Descrizione generale

Per l'esercitazione vengono utilizzati tre sistemi guest denominati **R1, H1, S1** collegati in rete tra loro. Sono host *Linux Debian 9.x*, su cui è possibile effettuare il **login** con i seguenti account: ***root***, ***user1***, ***user2*** tutti con password "***networks***". Per semplicità è possibile svolgere le esercitazioni effettuando l'accesso ai sistemi con le credenziali di *root*, ma al fine di attenersi alle buone pratiche si sconsiglia questo tipo di approccio, entrambi gli utenti *user1* e *user2* possono eseguire comandi con privilegi amministrativi invocando i programmi tramite "*sudo*", esempio:

```
$ sudo wireshark &
```

Nota: si ricorda che nelle linee di comando il carattere "*$*" o "*#*" ad inizio riga specifica la condizione di privilegio dell'utente: "*$*" corrisponde un utente con privilegi ristretti, mentre "*#*" indica il prompt dell'utente root.

La directory **C:\Temp** del sistema host fisico viene montata in automatico dai guest al mountpoint **/media/sf_temp** per permettere lo scambio di file tra i sistemi.



Si ricorda di effettuare sempre un copia di backup dei file di configurazione che si andranno a modificare.

La figura mostra la topologia di rete utilizzata.

Il router R1 permette il collegamento alla rete "reale" tramite il servizio NAT messo a disposizione da VirtualBox.

La seguente tabella illustra, per ogni interfaccia di rete dei sistemi guest, il nome definito da Virtualbox, il nome dell'interfaccia definito dal S.O., il MAC address associato e la rete a cui sono collegate.

	Interfaccia VB/S.O.	**MAC**	**LAN**
R1	Scheda1/enp0s3	aa:aa:aa:aa:00:00	VBox NAT
	Scheda2/enp0s8	aa:aa:aa:aa:00:11	LAN1
	Scheda3/enp0s9	aa:aa:aa:aa:00:22	LAN2
H1	Scheda1/enp0s3	cc:cc:cc:cc:00:00	LAN1
S1	Scheda1/enp0s3	ee:ee:ee:ee:00:00	LAN2

Dal sistema guest per ottenere una lista delle interfacce di rete eseguire

```
user1@r1:~$ ip link show
```

In Debian 9.x le interfacce di rete seguono uno schema di denominazione in base alla tipologia e alla loro connessione al sistema (integrata su scheda madre, su slot PCIex, ethernet, wireless…), nella configurazione utilizzata sono nella forma: ***enpXsY***[2] dove X,Y sono numeri che dipendono dalla configurazione del sistema e sono deducibili dall'output del comando *lspci*.

[2]Per maggiori dettagli si veda: https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/

ESERCITAZIONE 0

Si cerchi la configurazione di rete dell'host Windows su cui si sta lavorando (usare il comando `ipconfig`). Individuare l'indirizzo IP dell'host, la sua netmask, e l'indirizzo IP del default gateway. Si catturi, utilizzando opportuni filtri, il traffico generato dalla propria stazione a seguito del seguente comando digitato sulla stazione:

```
ping [indirizzo]
```

dove l'indirizzo da utilizzare è uno qualsiasi degli indirizzi IP di un altro PC del laboratorio.
Ci si accerti, prima di digitare il comando, che la stazione non abbia l'indirizzo MAC della macchina in esame nella propria cache ARP. Per controllare gli indirizzi memorizzati sulla macchina digitare:

```
arp -a
```

Per cancellare tutti gli indirizzi bisogna quindi eseguire:

```
arp2 -d
```

(Solitamente si utilizzerebbe il comando `arp -d` *con i privilegi di amministratore. Per problemi di privilegio sulle postazioni di laboratorio, per cancellare la cache ARP è necessario utilizzare il comando* `arp2` *da c:\windows al posto del comune arp.)*

Nota: il comando deve essere eseguito digitando un indirizzo IP (es. 10.1.1.1) e non un nome letterale (es. www.polito.it).

Domande

1. Trovata la configurazione di rete del host su cui si sta lavorando, indicare quale parte dell'indirizzo IP rappresenta la rete e quale l'host.
2. Analizzare la cattura e individuare eventuali pacchetti non relativi al comando digitato, che possono essere presenti se il filtro utilizzato non è sufficientemente restrittivo. Trascurare questi pacchetti nel rispondere alle successive domande e cercare eventualmente di nascondere tali pacchetti attraverso un opportuno filtro di visualizzazione di Wireshark.
3. Analizzare la cattura e indicare le funzionalità dei principali campi di un pacchetto IP (versione, indirizzo sorgente e destinazione, TTL, ToS). Attraverso quale campo dell'header si identifica il protocollo del payload contenuto nel pacchetto? Sono presenti pacchetti frammentati? Perché?
4. Quale coppia di pacchetti viene generata dall'esecuzione del comando `ping`? Indicare a chi appartengono gli indirizzi MAC sorgente e destinazione in essi contenuti.
5. MAC sorgente del secondo pacchetto e MAC destinatario del terzo pacchetto sono uguali? Perché?
6. Quale è l'indirizzo IP di destinazione nel primo pacchetto?
7. Che valore ha e a cosa serve il campo Target IP Address del primo pacchetto?

(Seconda Parte)

Utilizzando le stesse configurazioni esposte nella prima parte, ripetere l'esecuzione del comando ping utilizzando come target un indirizzo ip esterno alla rete del laboratorio.
Si catturi, utilizzando opportuni filtri, il traffico generato dalla propria stazione a seguito del seguente comando:

```
ping 8.8.8.8
```

Ci si accerti, prima di digitare il comando, di svuotare la cache ARP della stazione. (Comando `arp2` come descritto in precedenza).

Domande

1. Quali sono IP sorgente e IP destinazione dei pacchetti ICMP catturati?
2. Quali sono MAC sorgente e MAC destinazione dei pacchetti ICMP catturati?
3. A chi corrisponde l'indirizzo MAC destinazione dei pacchetti ICMP generati?
4. Indicare quali sono le principali differenze tra i pacchetti generati nella prima parte.
5. Indicare l'indirizzo MAC di 8.8.8.8

(Terza Parte)

Avviare ora la Virtual Machina *R1* e ripetere l'esecuzione del comando ping utilizzando come target un indirizzo ip esterno alla rete del laboratorio.
Si catturi, utilizzando opportuni filtri, il traffico generato dalla propria stazione a seguito del seguente comando:

```
ping -c 4 8.8.8.8
```

Per svuotare la cache ARP usare il comando `arp -d` eseguibile come root su Linux.

Domande

1. Quali sono IP sorgente e IP destinazione dei pacchetti ICMP catturati?
2. Quali sono MAC sorgente e MAC destinazione dei pacchetti ICMP catturati?
3. Quali sono le principali differenze tra i pacchetti generati nella seconda parte?

NOTA: Se lo studente lo ritiene opportuno, puó riportare nelle seguenti tabelle l'esito delle catture effettuate.

Cattura (Prima Parte)

```
ping [indirizzo ip laboratorio]
```

N.	L2	L3	Descrizione del Payload
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			

Cattura (Seconda Parte)

```
ping 8.8.8.8
```

N.	L2	L3	Descrizione del Payload
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			

Cattura (Terza Parte)

```
Ping -c 4 8.8.8.8
```

N.	L2	L3	Descrizione del Payload
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			

ESERCITAZIONE 1

Questa esercitazione prevede:

- studio dello spazio di indirizzamento di una semplice rete Ethernet dati una serie di requisiti
- la sua implementazione/configurazione a livello IP dei vari sistemi
- una breve analisi del traffico di rete

Avviare i tre nodi.

Prima parte

Dato il seguente range di indirizzi IP 192.168.0.0/16, si vuole realizzare una rete secondo la topologia data e con le seguenti specifiche:

- la connessione degli host alla LAN è fornita da uno switch
- LAN1 deve contenere circa 100 host
- LAN2 deve contenere circa 50 host
- Sia possibile aggiungere una terza LAN (LAN3) con ulteriori 50 host
- Utilizzare l'indirizzamento che garantisca il minor "spreco" di IP.
- Assegnare ai default gateway delle LAN l'ultimo IP disponibile della relativa sottorete.
- Assegnare agli host H1 e S1 il primo IP valido della relativa sottorete.
- R1 fornisce DHCP agli host di LAN1 e LAN2

Calcolare lo spazio di indirizzamento CIDR per le tre reti coinvolte e assegnare gli indirizzi IP ai sistemi guest descritti in precedenza

In totale occorre indirizzare circa 200 host, pertanto una rete in /24 del range a disposizione è sufficiente. Scegliamo ad esempio la prima disponibile ovvero 192.168.0.0/24 e suddividiamola nelle 3 LAN richieste. Per indirizzare i 100 host di LAN1 servono 7 bit nella parte di host quindi usiamo una rete in /25. Resta da suddividere il restante intervallo in due reti in grado di indirizzare 50 host ciascuna, sono necessari 6 bit per la parte di host, ovvero 2 reti in /26.

	Rete	Range	Netmask
LAN1	192.168.0.0/25	192.168.0.0 192.168.0.127	255.255.255.128
LAN2	192.168.0.128/26	192.168.0.128 192.168.0.191	255.255.255.192
LAN3	192.168.0.192/26	192.168.0.192 192.168.0.255	255.255.255.192

Assegnamento IP (CIDR)		
	NIC	**IP**
R1	enp0s3	10.0.2.15/24 (*VBox dhcp, NON modificare*)
R1	enp0s8	192.168.0.126/25
R1	enp0s9	192.168.0.190/26
H1	enp0s3	192.168.0.1/25
S1	enp0s3	192.168.0.129/26

Configurazione di rete degli host[3]

Il router R1 che fornisce il servizio DHCP avrà per le proprie interfacce una configurazione statica, mentre H1 e S1 andranno configurati in DHCP.
Le impostazioni di rete dei guest sono configurate tramite il file /etc/network/interfaces.
Editare[4] il file dei vari sistemi, ogni interfaccia di rete che si desidera configurare deve avere la propria sezione. Il semplice assegnamento di un IP (in modalità statica o dinamica) si ottiene specificando le seguenti impostazioni di **esempio**. In questa fase non configurare alcun gateway.

IP STATICO	**DHCP**
auto ***enp0s3*** iface ***enp0s3*** inet static address ***10.10.0.62*** netmask ***255.255.255.0***	auto ***enp0s3*** iface ***enp0s3*** inet dhcp

Nota: in grassetto-corsivo i dati da specificare in base alla propria configurazione
Per informazioni più dettagliate si rimanda alla *man page* del file (man interfaces)
Per applicare la nuova configurazione eseguire la seguente catena di comandi (specificare la corretta interfaccia di rete che si desidera aggiornare, in questo esempio "*enp0s8"*):

```
# ifdown -v enp0s8 && ip addr flush dev enp0s8 && ifup -v enp0s8
```

In alternativa, **prima**[5] di editare il file arrestare i servizi di rete con uno dei seguenti comandi:

```
# service networking stop
# systemctl stop networking.service
```

Al termine della modifica riavviare i servizi di rete con uno dei seguenti comandi:

```
# service networking restart
# systemctl restart networking.service
```

Per attivare/disattivare una singola interfaccia (ad es. la "*enp0s3*") eseguire

```
# ifup/ifdown enp0s3
```

Verificare la corretta configurazione con uno dei seguenti comandi:

```
# ip addr show   (consigliato)
# ifconfig
```

Affinché i client in dhcp ottengano i corretti parametri di rete occorre procedere alla configurazione del DHCP server

Configurazione DHCP server

Il servizio DHCP è in esecuzione su R1 e il relativo demone è ***dhcpd***. Editare il file di configurazione /etc/dhcp/dhcpd.conf aggiornandolo con i parametri opportuni: specificare i vari range di indirizzi IP coerenti con l'indirizzamento e gli assegnamenti statici scelti in precedenza e attribuire a H1 e S1 una "*reservation*" in modo che il DHCP server fornisca a tali sistemi sempre il medesimo IP, porre attenzione ai valori dei MAC address da specificare affinché la reservation su H1 e S1 sia correttamente configurata.
Usare la man page per ulteriori informazioni (man dhcpd.conf).
Nota: eventuali reservation specificate nel file di configurazione del dhcp ed eventuali indirizzi configurati staticamente sui client **non devono** essere inclusi nei range di IP serviti dal server dhcp.
Riavviare il servizio invocando uno dei seguenti comandi:

```
# service isc-dhcp-server restart
# systemctl restart isc-dhcp-server.service
```

[3]Esistono svariati modi per configurare le interfacce di rete oltre a quelli illustrati nella seguente documentazione.
[4]Per editare il file occorrono i privilegi amministrativi. Sono disponibili i seguenti editor: vim, nano, leafpad.
[5]In caso contrario il vecchio IP può non essere eliminato ottenendo una scorretta configurazione di rete e solo con il comando "ip addr show" è possibile osservare l'assegnamento di due indirizzi IP all'interfaccia.

Se non vengono visualizzati messaggi il comando è andato a buon fine, la sintassi del file di configurazione è corretta e il servizio dhcp attivo.
Il server DHCP scrive i propri messaggi di log nel file "*/var/log/syslog"* , in caso di problemi visualizzare a video tale file con uno dei seguenti comandi

```
# cat /var/log/syslog
# tail [-f] /var/log/syslog
```

Il comando **netstat** permette di visualizzare informazioni riguardo alle connessioni di rete.
Verificare che su R1 sia attivo il servizio dhcp identificando indirizzi, protocolli e porte utilizzati[6] dal demone *dhcpd*

```
# netstat -anup |grep dhcpd
udp      0      0 0.0.0.0:12165           0.0.0.0:*               1938/dhcpd
udp      0      0 0.0.0.0:67              0.0.0.0:*               1938/dhcpd
udp6     0      0 :::53949                :::*                    1938/dhcpd
```

Verifica della configurazione

Per assicurarsi che i client abbiano la corretta configurazione di rete riavviare (sui client) il servizio *networking* e verificare con il comando ifconfig che la configurazione sia quella desiderata. In alternativa è possibile utilizzare:

```
# ip addr show
```

Nel caso vengano visualizzati più indirizzi IP per interfaccia occorre procedere alla rimozione dell'IP non utilizzato. Esempio rimozione di IP *192.168.0.180* dall'interfaccia *enp0s9*:

```
# ip addr del 192.168.0.180/26 dev enp0s9
```

Analisi di rete

Prima di procedere assicurarsi che la cache arp sia vuota, questo permette di visualizzare effettivamente tutto il traffico che verrà generato. Sui nodi eseguire da root :

```
# arp -a
```

eventualmente eliminare le voci presenti con

```
# arp -d N.N.N.N
```

dove *N.N.N.N* rappresenta l'indirizzo IP della voce arp che si desidera eliminare.
Su R1 avviare (con diritti amministrativi) Wireshark per catturare il solo traffico sulle interfacce di rete attestate alle due LAN, effettuare un ping verso H1 e S1 specificandone gli indirizzi IP

```
$ ping 192.168.x.x
```

1. Quali coppie di pacchetti vengono generate dall'esecuzione del comando ping? Indicare a chi appartengono gli indirizzi MAC sorgente e destinazione in essi contenuti.

Vengono generate secondo questa sequenza le seguenti due coppie di trame

1.	ARP request:	MAC src=R1	MAC dst=H1|S1
2.	ARP reply:	MAC src=H1|S1	MAC dst=R1
e			
3.	ECHO request:	MAC src=R1	MAC dst=H1|S1
4.	ECHO reply:	MAC src=H1|S1	MAC dst=R1

La prima coppia permette ai due nodi di scambiarsi le informazioni relative al proprio MAC address.
La seconda coppia è il pacchetto ICMP e si ripete un numero arbitrario di volte.

2. Per quale motivo sono presenti altri pacchetti oltre a quelli del protocollo icmp?

[6]La presenza di due porte UDP casuali oltre alla 67 (definita dalle specifiche di protocollo) è dovuta alle opzioni specificate in fase di compilazione del programma e sono relative a funzionalità DDNS (vedere man dhcpd)

Assumendo che la cache arp del nodo sorgente sia vuota, prima della richiesta echo reply viene generata una arp request broadcast per risalire al MAC address dell'host destinatario.

Al termine del ping anche l'host destinazione effettua una arp request, questa volta unicast, verso il MAC da cui ha ricevuto la richiesta di echo, come refresh e validazione della propria cache arp

3. MAC sorgente della seconda trama e MAC destinatario della terza trama sono uguali? Perché?

Con la seconda trama l'host destinatario comunica, al sorgente della richiesta di echo, il proprio MAC. Il nodo sorgente può ora comporre la trama in cui imbustare il pacchetto IP+ICMP e inviarlo sulla rete. (terza trama).

4. Effettuare un ping tra H1 e S1? Quale può essere la causa della mancata connessione?

Comunicano solo le coppie R1-H1 e R1-S1 perché sui sistemi non è ancora presente una tabella di routing.

5. Analizzare la cattura e indicare le funzionalità dei principali campi di un datagram IP (versione, indirizzo sorgente e destinazione, TTL, ToS). Attraverso quale campo dell'header si identifica il protocollo del payload contenuto nel datagram? Sono presenti datagram frammentati? Perché?

Si consulti il libro di testo del corso o RFC 791 e successivi aggiornamenti https://tools.ietf.org/html/rfc791

Il campo PROTOCOL identifica il protocollo del payload.

Non sono presenti datagram frammentati perché la dimensione complessiva di header IP e payload (protocollo ICMP in questo esempio) è inferiore alla MTU di Ethernet (1500 bytes).

6. Catturare il traffico di rete generato da una richiesta DHCP fermando e riavviando i servizi di rete su H1 o S1. Si raccomanda di lanciare la cattura su sull'altra macchina rispetto a quella dove i servizi vengono riavviati.

Quali IP (sorgente e destinazione) vengono utilizzati per avviare la comunicazione? Quale protocollo di trasporto viene utilizzato? Specificare le porte sorgente e destinazione del primo datagram.

Quante trame sono necessarie per portare a termine l'operazione?

```
# service networking stop
# service networking start
```

Il client invia una trama broadcast con un indirizzo IP sorgente 0.0.0.0 e IP destinazione 255.255.255.255. Il protocollo di trasporto è UDP con porta sorgente 68 e destinazione 67.

Per portare a termine l'assegnamento dell'indirizzo sono necessarie 4 trame.

Seconda parte

Configurazione delle tabelle di routing

Visualizzare le correnti tabelle di routing dei tre nodi invocando

```
$ ip route show
```

oppure da root

```
# route -n
```

1. Quali host hanno la necessità di configurare la propria routing table? Perché

Solo H1 e S1 devono configurare la tabella con il valore del default gateway.

R1 infatti conosce tutte le reti a cui è connesso e il default gateway per la coonnessione verso l'esterno gli viene fornito dal dhcp della rete NAT di VirtualBox. Nella nostra configurazione il DHCP in esecuzione su R fornisce ai client solo l'indirizzo IP, pertanto occorre procedere manualmente alla configurazione del default gateway e della tabella di routing .

Per aggiungere una entry nella tabella di instradamento occorre eseguire[7]:

```
# ip route add IP_NETWORK/XX via IP_GW dev enpXsY
```

- *IP_NETWORK/XX* : la rete da raggiungere nel formato CIDR

[7]In alternativa è possibile utilizzare anche il comando 'route', si veda la man page per la sua sintassi e dettagli.

- *IP_GW* : IP del gateway attraverso cui raggiungere la rete specificata
- *enpXsY* : interfaccia di rete attraverso cui il GW è raggiungibile

Configurare le tabelle dei nodi H1 e S1 per permettere l'instradamento nella rete interna

H1:	# ip route add 192.168.0.128/26 via 192.168.0.126 dev enp0s3
S1:	# ip route add 192.168.0.0/25 via 192.168.0.190 dev enp0s3

Eseguire **una sola** richiesta ping (con l'opzione "*-c 1*") tra H1 e S1 e catturare il traffico in transito su R1 selezionando le interfacce di rete *enp0s8*, *enp0s9*.

2. Perché nonostante l'opzione "*-c 1*" vengono visualizzati due pacchetti di *echo request* e due di *echo reply*? Elencare i motivi di tale comportamento e le differenze tra frame dello stesso tipo catturati

La prima trama è catturata sulla enp0s8: è la consegna diretta che H1 ha eseguito verso il proprio next hop, il TTL è pari a 64, la trama ethernet ha MAC sorgente di H1 e destinazione di R1, inoltre ha un proprio valore di checksum.

La seconda trama è catturata sulla *enp0s9*: dopo aver analizzato la propria tabella di instradamento R1 effettua la consegna diretta del pacchetto al destinatario S1, il TTL è decrementato di 1 e vale 63, la trama ethernet ha MAC sorgente di R1 e destinazione di S1, inoltre ha un proprio valore per il checksum.

Simili considerazioni valgono per le trame di risposta *echo reply.*

Eseguire da H1 con privilegi di root il seguente comando e catturare il traffico in transito sulle interfacce *enp0s8*, *enp0s9* di R1

```
# traceroute -I -q 1 192.168.0.129
```

3. Analizzare la cattura e descrivere il funzionamento del comando.

traceroute cerca di tracciare il percorso che un pacchetto segue per arrivare alla destinazione specificata, elencando i vari router che attraversa. Per ottenere questo risultato il programma effettua una serie di ping verso la destinazione impostando per ogni pacchetto un TTL crescente partendo dal valore 1. Al primo pacchetto icmp inviato il next hop (primo router incontrato ovvero il default gateway) decrementa il TTL che va a 0, scarta il datagram e se è configurato opportunamente risponde con un messaggio icmp di tipo 11 (Time To Live exceed). L'host sorgente invia una nuova richiesta di echo incrementando di 1 il TTL e permettendo al pacchetto di raggiungere il secondo next hop che scarterà a sua volta il datagram (avendo TTL=0) e risponderà con un nuovo icmp tipo 11, e così via fino ad arrivare alla destinazione.

Verificare che il traffico dei due sistemi H1 e S1 non viene instradato verso ulteriori destinazioni oltre alle LAN1 e LAN2 effettuando un ping a siti esterni (ad es. 8.8.8.8)
Aggiornare le tabelle di routing dei nodi H1 e S1 per permettere l'instradamento anche verso l'esterno (suggerimento: la generica destinazione è denominata "*default*")

H1:	# ip route add default via 192.168.0.126 dev enp0s3
S1:	# ip route add default via 192.168.0.190 dev enp0s3

Visualizzare la nuova tabella di routing e verificare l'effettiva connettività verso l'esterno.

4. È possibile semplificare la tabella di routing ottenuta?

Dato che H1 e S1 hanno una sola interfacccia di rete è possibile specificare un'unica route di default per tutte le destinazioni, pertanto gli instradamenti verso LAN1/LAN2 sono superflui.

5. Ripetere il comando *traceroute* usando come destinazione 8.8.8.8 e catturare il traffico generato
   - Tutti i router attraversati rispondono con un icmp di tipo 11 (TTL exceed)?

No, non tutti i router sono configurati per rispondere con pacchetti ICMP TTL exceed

Aggiornamento configurazione DHCP server

Per fornire ai client la corretta tabella di routing occorre aggiornare il file di configurazione del server DHCP e specificare per le sottoreti servite il loro *default gateway* : aggiungere la seguente direttiva, con l'opportuno valore, nelle sezioni *subnet:*

```
option routers 192.168.x.x;
```

Terza parte: una rete reale

Effettuare una cattura direttamente dall'host del laboratorio in modo da osservare il comportamento di una rete reale.
Cosa cambia rispetto alla situazione simulata per l'esercitazione?

In una rete reale sono in esecuzione su molti host molti servizi (come ad es. autenticazione e connessioni a dischi di rete) che per il loro corretto funzionamento generano in continuazione traffico sulla rete.

Senza osservare la configurazione di rete ma solo con la cattura attraverso Wireshark ricavare il default gateway e il DNS dell'host del laboratorio.

Senza conoscere la rete del laboratorio è possibile fare un ping a *www.polito.it,* che si trova in una rete esterna alla LAN, in modo che il nodo effettui una richiesta DNS per la risoluzione della destinazione e successivamente invii la trama ethernet con la richiesta echo al proprio default gateway.

ESERCITAZIONE 2

Questa esercitazione vuole illustrare il funzionamento di un semplice sistema DNS e prevede:

- di associare alla rete in esame uno spazio dei nomi
- di configurare per tale spazio un server dei nomi assoluto (detto anche autoritativo) in grado di inoltrare a un server esterno le query che non è in grado di soddisfare
- breve analisi del traffico

Avviare i tre nodi.

Prima parte

DNS[8]: panoramica

Un server DNS può essere configurato in diversi modi:

- **server autoritativo primario** : fornisce e detiene le informazioni di zona/e per cui è autoritativo, può pertanto apportare modifiche al database DNS per le zone di propria competenza;
- **server autoritativo secondario**: fornisce le informazioni di zona/e ottenute da da un altro server autoritativo per quella zona/e, in pratica è una replica di un altro server assoluto;
- **server cache**: risolve le interrogazioni per conto degli hosts della rete e detiene in cache una copia dei risultati ottenuti per evadere successive richieste, non possiede alcuna competenza di zona.

Una zona rappresenta lo spazio dei nomi (dominio) gestito da uno o più server (un primario e uno o più secondari) che vengono definiti autoritativi per quella zona.
Il sistema DNS memorizza le informazioni in cosidetti *record di risorsa* o *resource record* **RR**, costituiti nella loro forma elementare da tre campi: '*Name*, *Type*, *Value*' ('*Nome*, *Tipo*, *Valore*') . I campi *Name* e *Value* assumono un determinato significato in base al contenuto del campo *Type*.
Ad esempio un record di tipo *A* associa alla stringa alfanumerica del campo *Name* l'indirizzo IP del campo *Value* permettendone la traduzione:

Name	*Type*	*Value*
server01.polito.it.	A	192.168.10.10

Un record di tipo *CNAME* associa alla stringa del campo *Name* il valore contenuto nel campo *Name* di un record di tipo *A*, ovvero fornisce un alias per un determinato hostname.

Name	*Type*	*Value*
www.polito.it.	CNAME	server01.polito.it.

Il DNS oltre a fornire il servizio di traduzione da nome di dominio a indirizzo IP permette anche la cosiddetta ***risoluzione inversa***, cioè dato un IP consente di risalire al suo hostname. In questo caso gli indirizzi IP vengono trattati come "etichette" ovvero sono considerati come *nomi* di un apposito dominio, denominato ***in-addr.arpa,*** ai quali vengono associati record di tipo **PTR** (puntatori a un'altra parte dello spazio dei nomi) il cui valore è l'hostname ricercato. Un record di tipo PTR ha come valore del campo *Value* il contenuto del campo *Name* di un record di tipo A. I nomi di dominio della zona *in-addr.arpa* rappresentano gli IP che si desidera tradurre specificati però con gli ottetti in ordine inverso.

[8]Per maggiori dettagli si consultino RFC 1034,1035 e successivi.
Per un elenco di RFC relativo a DNS si consulti https://www.isc.org/community/rfcs/dns/

Ad esempio nel dominio di ricerca inversa il nome associato all'IP "*192.168.0.1"* sarà "*1.0.168.192.in-addr.arpa"* e il relativo record di tipo *PTR* sarà

```
            Name                        Type           Value
1.0.168.192.in-addr.arpa.               PTR         nome-host.nome.del.dominio.
```

a cui corrisponde nel dominio "*nome.del.dominio*" il seguente record di tipo *A*

```
            Name                        Type           Value
nome-host.nome.del.dominio.              A          192.168.0.1
```

Nota: fare attenzione al '.' (punto) con cui terminano tutti i valori del campo *Name*.

La zona *in.addr.arpa* è a tutti gli effetti un dominio DNS come gli altri e come tale viene trattato dal sistema DNS, è grazie all'uso della notazione inversa che si può risalire al record tipo A pertinente l'IP dell'interrogazione.

Per la risoluzione inversa degli IP della rete 192.168.0.0/24 occorre configurare nel DNS la zona "inversa" *0.168.192.in-addr.arpa* e popolarla con i record PTR che si desidera tradurre.

Implementazione di un servizio DNS

Il server DNS della distribuzione Debian 9.x è BIND9[9] (ver. 9.10.3) di Internet Systems Consortium (ISC), uno dei più diffusi server DNS.

Alla rete configurata nella prima esercitazione si vuole associare uno spazio dei nomi. La rete viene identificata dal dominio "**corp.local**" per cui S1 è il server dei nomi assoluto.

In base alla seguente tabella che illustra IP, nomi e servizi assegnati ai vari nodi, determinare i tipi di **record di risorse** per il popolamento del DNS. Identificare anche le zone di risoluzione inversa con i relativi record di tipo PTR.

IP	Nome DNS	SERVIZIO
10.0.2.15 192.168.0.126 192.168.0.190	r1.corp.local	
192.168.0.1	h1.corp.local	
192.168.0.129	s1.corp.local www.corp.local intra.corp.local corp.local mail.corp.local pop3.corp.local	 web web mail (invio/inoltro) mail (invio/inoltro) mail (lettura)

Note e suggerimenti.

Tutti gli IP di una cella vanno associati tramite l'opportuno tipo di record ai valori presenti nella cella adiacente (colonna *Nome DNS*).

La colonna *SERVIZIO* indica i servizi associati al *Nome DNS* della riga corrispondente.

E' possibile associare a un nome DNS più record di tipo A (ad es. nel caso di sistemi multihomed).

Ogni record di tipo A deve avere il corrispondente record PTR nel dominio di risoluzione inversa *in-addr.arpa*.

S1 è server autoritativo (ogni zona contiene un record NS che specifica il proprio server assoluto):
per la zona: *corp.local*
per le seguenti zone di ricerca inversa: *0.168.192.in-addr.arpa* *2.0.10.in-addr.arpa*

Un tipo MX *deve* essere associato a un record di tipo A, ovvero il campo VALUE deve contenere una stringa corrispondente al campo NAME di un record di tipo A e comunque mai contenere il nome di un record tipo CNAME[10].

[9]https://www.isc.org/downloads/bind/ per software e manuali o sul sito Ladispe sez. *Corsi - Reti di Calcolatori*

[10]Si consulti RFC 2181 sezione 10.3 (https://tools.ietf.org/html/rfc2181#section-10)

Il campo *Name* di un record tipo CNAME *non può e non deve coesistere*[11] con altri tipi di record, ovvero il suo contenuto deve essere univoco: non deve esistere ad esempio un record tipo CNAME che abbia nel proprio campo *Name* lo stesso valore del campo *Name* di un record tipo NS.

RR per zona ***corp.local***		
Name	**Type**	**Value**
corp.local.	NS	s1.corp.local.
corp.local.	MX	mail.corp.local.
r1.corp.local.	A	10.0.2.15
	A	192.168.0.126
	A	192.168.0.190
s1.corp.local.	A	192.168.0.129
mail.corp.local.	A	192.168.0.129
pop3.corp.local.	CNAME	s1.corp.local.
www.corp.local.	CNAME	s1.corp.local.
intra.corp.local.	CNAME	s1.corp.local.
h1.corp.local.	A	192.168.0.1

RR per zona ***0.168.192.in-addr.arpa***		
Name	**Type**	**Value**
0.168.192.in-addr.arpa.	NS	s1.corp.local.
1.0.168.192.in-addr.arpa.	PTR	h1.rd.corp.local.
129.0.168.192.in-addr.arpa.	PTR	s1.corp.local.
129.0.168.192.in-addr.arpa.	PTR	mail.corp.local.
126.0.168.192.in-addr.arpa.	PTR	r1.corp.local.
190.0.168.192.in-addr.arpa.	PTR	r1.corp.local.
RR per zona ***2.0.10.in-addr.arpa***		
2.0.10.in-addr.arpa.	NS	s1.corp.local.
15.2.0.10.in-addr.arpa.	PTR	r1.corp.local.

[11]Si consulti RFC 1912 sezione 2.4 (https://tools.ietf.org/html/rfc1912#section-2.4)

Configurazione BIND9 su S1

La configurazione di BIND è costituita da una serie di file, solitamente in /etc/bind/ , che vengono inclusi dal file di configurazione principale named.conf :
- named.conf.options: vengono specificate le opzioni generali come directory di lavoro, interfacce e porte di ascolto per il servizio;
- named.conf.local: contiene la configurazione delle zone gestite dal server;
- named.conf.default-zones: contiene configurazione per alcune zone di default come i root-server;
- db.*: file di zona in cui vengono specificati i RR.

Editare named.conf.options:

- aggiungere come IP di ascolto quello associato all'interfaccia di rete e quello di loopback specificando la seguente direttiva:

```
listen-on { ip1; ip2; };
```

- abilitare la possibilità di inoltro delle query per le richieste provenienti da qualsiasi host con le direttive (una configurazione di produzione DEVE essere più restrittiva):

```
recursion yes;
allow-recursion { any; };
```

- configurare come DNS di forward *130.192.225.79* e permettere l'inoltro delle query solo a tale server tramite la direttive*:*

```
forward only;
forwarders { ip; };
```

- disabilitare l'ascolto su IPv6 specificando *none* nella direttiva *listen-on-v6*
- commentare la direttiva *dnnsec-validation*

Editare named.conf.local:

- definire le zone per cui **S1 è autoritativo** (zone di ricerca sia diretta che inversa) e specificare i file contenenti i RR delle zone. Adattare il seguente esempio alla configurazione da realizzare.

```
zone "nome.di.zona" {
        // The server has a master copy of the data for the zone
        // and will be able to provide authoritative answers for it
        type master;
        // file with RRs
        file "/etc/bind/db.nome.di.zona";
};
zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.1.168.192.in-addr";
};
```

Creare i file di zona *db.** specificati nel precedente file e popolarli con i RR opportuni. Adattare il seguente esempio alla configurazione da realizzare, per tutte e tre le zone da configurare.

```
$TTL    86400
@          IN      SOA       dns.example.org. root.localhost. (
                                  1             ; Serial
                                  604800    ; Refresh
                                  86400         ; Retry
                                  2419200           ; Expire
                                  86400 )       ; Negative Cache TTL
```

```
                        IN      NS      dns.example.org.
                        IN      MX      10 mail.example.org.
dns                     IN      A       192.168.1.1
                        IN      A       192.168.10.1
                        IN      A       10.0.1.1
mail                    IN      A       192.168.3.3
web01.example.org.      IN      A       192.168.3.2
                        IN      HINFO   Intel_Xeon
www.example.org.        IN      CNAME   web01
```

Note e suggerimenti[12].

Il primo simbolo "@" rappresenta il nome della zona come definito nel file *named.conf.local*.

Il record di tipo SOA (Start of Authority), il cui nome corrisponde a quello di zona (perché la prima colonna è "@"), restituisce informazioni generali sulla zona DNS. In questo esempio la zona è *example.org*, *dns.example.org.* è il server DNS principale e *root.localhost* è l'e-mail dell'amministratore (dove la "@" è sostituita da "."); vi sono poi il numero seriale del dominio (da aggiornare ad ogni modifica apportata alla zona in modo che i server secondari si accorgano del cambiamento) e diversi timer che regolano la frequenza di trasferimento e la durata di validità dei record.

I nomi host che non terminano con un "." (punto) vengono trattati come nomi relativi (non FQDN) a cui viene automaticamente aggiunto come suffisso il nome di zona.

Le righe che iniziano con uno spazio bianco o con "@" fanno riferimento al record precedente: in questo esempio a *dns.example.org.* sono associati tre record di tipo A.

Il valore "IN" nella seconda colonna specifica la classe del record (IN=internet).

Il valore "10" dopo il campo MX rappresenta un valore di priorità utilizzato nel caso esistano più record MX.

Aggiungere inoltre per ogni hostname (quindi RR tipo 'A') i seguenti tipi di RR:

- HINFO per informazioni su CPU e SO (RFC 1035) es:

```
IN      HINFO       vCPU Debian
```

- TXT campo testo generico (RFC 1035) es:

```
IN        TXT       testo_descrittivo
```

Riavviare il servizio con

```
# systemctl restart bind9.service
```

Verificare l'assenza di errori

```
# systemctl status bind9.service
```

Verificare con *netstat* che il servizio sia in ascolto sulle interfacce di rete prestabilite.

Aggiornamento DHCP su R1

Occorre aggiornare la configurazione e i parametri del server DHCP per permettere ai client, che utilizzano tale servizio, di ottenere la corretta configurazione relativa al DNS. Sul server DHCP editare il file `/etc/dhcp/dhcpd.conf` e specificare nella sezione globale (fuori da ogni dichiarazione *subnet* o *group*) le seguenti direttive con i corretti valori:

```
option domain-name "nome.dominio.";
option domain-name-servers IP_SERVER_DNS;
```

Aggiungere anche nella sezione globale la seguente opzione che permette di indicare una lista di domini da "appendere" ai nomi host nel caso non venisse fornito il FQDN (ad esempio in un comando *ping*)

```
 option domain-search "nome.dominio.";
```

Riavviare il servizio per rendere effettive le modifiche.

[12]Per maggiori dettagli si consultino RFC 1034,1035 e successivi.

Il sistema R1 è già configurato correttamente per l'utilizzo di S1 come DNS[13].

Seconda parte

Tool per DNS

Per interrogare un DNS in ambiente Unix-like si fa uso del comando "***dig***" o "***host***" mentre in ambito Windows si utilizza "***nslookup***". Anche in ambienti Unix-like è presente *nslookup* ma può avere comportamenti non consistenti e non uniformi tra diverse distribuzioni, pertanto se ne sconsiglia l'utilizzo.

dig (domain information groper): breve panoramica

```
dig  [@server] [-b address] [-c class] [-f filename] [-k filename] [-m]
     [-p port#] [-q name] [-t type] [-x addr] [-y [hmac:]name:key] [-4] [-6]
     { [name] [type] [class] } [queryopt...]
```

Nella forma più semplice si usa la seguente sintassi

```
dig @server name type
```

- **server** è l'hostname o l'IP del server DNS da interrogare
- **name** è il nome del record che si desidera ricercare (ad es. www.example.com)
- **type** è il tipo di record dell'interrogazione (ad es. A, CNAME, MX)

Se "*@server*" viene omesso *dig* consulta il file /etc/resolv.conf per trovare un DNS server.
Se "*type*" è omesso viene usato di default il tipo "A"

Ad esempio per trovare l'IP associato all'host www.example.com

```
dig @IP.SERVER.DNS www.example.com A
```

Di default le risposte sono molto prolisse e vengono visualizzate anche informazioni riguardanti la query effettuata e il server autoritativo per la zona

Opzioni di interrogazione uso più comune:

- **-q**: specifica il nome del record da ricercare.
- **-t**: specifica il tipo di record. Se *-q* è impostato a un nome di dominio e *-t* vale "AXFR" viene richiesto un trasferimento di zona.
- **-x**: specifica l'indirizzo IP di cui fare il *reverse lookup*. In questo caso non sono richieste le opzioni *-q* e *-t*, dig effettua di default una query impostando come nome la notazione IP inversa con suffisso *.in-addr.arpa* e come tipo *PTR*

Ripetendo i precedenti parametri è possibile effettuare interrogazioni su più record.

Query options sono opzioni generali che influenzano le interrogazioni richieste e la visualizzazione delle risposte. Ogni opzione è preceduta da un "+" ed eventualmente da "*no*" per disabilitarne l'effetto. Se vengono specificate subito dopo il comando *dig* assumono validità globale per tutte le interrogazioni

- **+[no]tcp:** [non]utilizza il protocollo TCP quando il tipo di interrogazione vale AXFR.
- **+[no]recurse:** [non]imposta l'interrogazione ricorsiva (default ricorsiva).
- **+[*no*]trace:** [non]imposta l'interrogazione come iterativa. Se abilitata dig effettua una query iterativa a partire dal root server (TLD) del nome ricercato e visualizza le risposte di ogni server intermedio interrogato per risolvere la query (default disabilitata).
- **+[*no*]short:** [non]imposta l'output verboso che è il valore usato di default.

Altre opzioni utili possono essere le seguenti:

- +[no]cmd

[13]La configurazione è stata forzata "staticamente" tramite il file */etc/dhcp/dhclient.conf*

- +[no]comments
- +[no]question
- +[no]all
- +[no]answer

Esempi

```
dig @8.8.8.8 www.google.it A   (restituisce l'IP di www.google.it)
dig +ncmd +nocomments +noquestion @8.8.8.8 www.google.it A   (come precedente)
dig @8.8.4.4 google.it NS   (restituisce i DNS autoritativi per il dominio google.it)
dig @8.8.4.4 -x 8.8.8.8   (effettua la ricerca inversa per l'IP 8.8.8.8)
dig @8.8.8.8 google.it -t AXFR   (richiede un trasferimento di zona)
```

Nota: generalmente i server DNS consentono il trasferimento di zona a pochi host fidati

rnd: name server control utility

Controlla le operazioni del server, come ad esempio:

- cancellazione della cache

```
rndc flush
```

- ricaricare file e zone di configurazione dopo una modifica

```
rndc reload
```

Per tutti i comandi illustrati si rimanda alla man page per una descrizione completa.

Interrogazioni DNS

Su H1 avviare Wireshark ed eseguire le seguenti interrogazioni selezionando S1 come server DNS.

1. Verificare che *h1.corp.local* è un record di tipo A.

```
dig h1.corp.local A
```

2. Visualizzare i record associati agli IP *192.168.0.129* e *10.0.2.15*

```
dig -x 192.168.0.129 -x 10.0.2.15
```

3. Verificare che *www.corp.local* è un alias e determinare il record di tipo A cui fa riferimento

```
dig -q www.corp.local -t CNAME
dig -q www.corp.local -t A
```

4. Visualizzare tutti i record associati all'host *r1.corp.local*

```
dig -q r1.corp.local -t ANY
```

5. Visualizzare tutti i record associati a *corp.local*

```
dig -q corp.local -t ANY
```

6. Visualizzare tutti i record della zona *corp.local*

```
dig -q corp.local -t AXFR
```

   - Cosa varia nella comunicazione tra client e server rispetto alle interrogazioni precedenti? Provare a indicarne la motivazione.

Nelle interrogazioni precedenti su record singoli viene utilizzato il protocollo UDP, mentre per i trasferimenti di zona (effettuati solitamente tra server DNS per aggiornamento della configurazione di zona) si utilizza TCP in modo da avere un trasferimento affidabile dei dati.

7. Effettuare un query non ricorsiva per ricavare il record di tipo A associato a *ipv6.polito.it*. Che risposta si ottiene? Perché?

```
dig -q ipv6.polito.it -t A +norecurse
```

Dato che S1 non è in grado di risolvere *ipv6.polito.it* (non è un server autoritativo per la zona *polito.it* , inoltre non ha l'informazione nella propria cache) e che la query è non ricorsiva, BIND non inoltra la richiesta al forwarder configurato, ma risponde con un elenco di root server, gli unici in base alla propria configurazione capaci di evadere la richiesta.

8. Ripetere la query richiedendo al server la ricorsione. Cosa si ottiene? Verificare il comportamento ricorsivo del server DNS eseguendo Wireshark su S1 per osservare il traffico generato.

Si ottengono le informazioni richieste. Il server inoltra la richiesta al proprio forwarder

9. Ripetere il punto 7. Si ottiene risposta ora? Perché?

Si ottiene la risposta come al punto 8 precedente perché ora il DNS ha in cache le informazioni richieste.

Terza parte (Facoltativa)

Spostare ora l'host *h1* in un nuovo sottodominio, aggiungere una nuova zona DNS denominata *rd.corp.local* in cui registrare l'host H1 con il suo vero hostname "*H1"*. In questo modo, l'host sarà identificato come *h1.rd.corp.local*. Inoltre si desidera che *rd.corp.local* rappresenti anche un host all'indirizzo *192.168.0.130* raggiungibile anche con *www.rd.corp.local*
Determinare i record necessari per la nuova zona e riconfigurare opportunamente il sistema in modo da permettere sia la ricerca diretta che inversa.

Suggerimenti.

- Nella zona *corp.local* eliminare i RR associati a *h1.rd.corp.local*
- Aggiornare il file *named.conf.local* aggiungendo la nuova zona
- Creare un nuovo file di zona *db.rd.corp.local* con i necessari RR (SOA, NS, A, CNAME, HINFO, TXT)
- Aggiornare il file di zona *0.168.192.in-addr.arpa* aggiungendo il nuovo record PTR

Modifica RR per zona ***corp.local***		
Name	**Type**	**Value**
~~h1.corp.local~~	~~A~~	~~192.168.0.1~~

RR per nuova zona ***rd.corp.local***		
Name	**Type**	**Value**
rd.corp.local.	NS	s1.corp.local
	A	192.168.0.130
h1.rd.corp.local.	A	192.168.0.1
www.rd.corp.local.	CNAME	rd.corp.local

Nuovo RR per zona ***0.168.192.in-addr.arpa***		
Name	**Type**	**Value**
130.0.168.192.in-addr.arpa.	PTR	rd.corp.local

Ricaricare la configurazione di BIND

```
# rndc reload
```

1. Effettuare un'interrogazione di tipo A e di tipo NS per il record *rd.corp.local* per verificare la corretta traduzione.

```
dig -q rd.corp.local -t A
dig -q rd.corp.local -t NS
```

2. Effettuare un'interrogazione per tutti i record associati a *rd.corp.local*

```
dig -q rd.corp.local -t ANY
```

3. Effettuare un trasferimento della zona *rd.corp.local* per visualizzarne tutti i record

```
dig -t AXFR rd.corp.local
```

4. Ripetere il punto 2 specificando sia come server DNS che come valore di interrogazione *ipv6.polito.it*.

```
dig @dns.ipv6.polito.it -q ipv6.polito.it -t ANY
```

   - Cosa si può dire riguardo al record appena cercato? (numero di server autoritativi, mail server del dominio, indirizzo mail dell'amministratore...)

ipv6.polito.it rappresenta un nome di dominio con 4 server autoritativi (record SOA e NS) mentre *dns.ipv6.polito.it* è un hostname (record A). Inoltre è il mail server autorizzato ad accettare messaggi mail per il dominio (record MX). La mail dell'amministratore è *risso@polito.it*

5. Richiedere un trasferimento per la zona *ipv6.polito.it* al server *130.192.225.79*

```
dig @dns.ipv6.polito.it -q ipv6.polito.it -t AXFR
```

Cosa cambia tra un'interrogazione tipo ANY e una di tipo AXFR effettuate entrambe su un nome di zona?

L'interrogazione di tipo ANY visualizza i record associati al nome richiesto (sia esso di zona o un hostname) e usa UDP. L'interrogazione AXFR richiede un trasferimento di zona permettendo di visualizzare tutti i record della zona e usa TCP, se il nome non è un nome di zona viene visualizzato un errore.

In base alla rete da cui ci si collega (Ladispe o altra rete esterna) si ottengono output differenti: quale può essere la ragione di tale comportamento?

In genere i trasferimenti di zona sono molto onerosi e vengono consentiti solo verso server e reti fidate. In questo caso il server interrogato permette i trasferimenti di zona solo verso la rete Ladispe e server interni alla rete del Politecnico, pertanto ogni richiesta proveniente da reti "sconosciute" ovvero non fidate viene negata.

ESERCITAZIONE 3

Questa esercitazione prevede:

- la configurazione sul medesimo server web di due siti aventi hostname differenti sfruttando il campo "*Host*" dell'header *HTTP/1.1* (name based Virtual Hosting)
- invio e lettura di mail tramite *telnet* con il protocollo *SMTP* e *POP3*
- breve analisi del traffico

Avviare i tre nodi: R1 fornisce DHCP e routing, S1 server web e posta, H1 client generico da cui effettuare le varie operazioni richieste.

Prima parte

Il server web da configurare è *Apache 2.4.25* sul sistema guest S1.
I due siti corrispondono agli hostname configurati per l'esercitazione 2 ovvero *www.corp.local* e *intra.corp.local* , i cui file sono già memorizzati in due directory all'interno di /var/www/
Occorre configurare correttamente il server web affinché le richieste vengano indirizzate ai siti opportuni.
Apache utilizza la direttiva *VirtualHost*[14] che permette di specificare a quale sito redirigere la richiesta in base all'header "*Host*" di HTTP/1.1.
In Debian l'elenco dei siti configurati nel sistema e disponibili alla pubblicazione on-line si trova in /etc/apache2/sites-available. Creare in questa cartella due copie del file di base 000-default.conf nominandole ad es. www.corp.conf e intra.corp.conf che saranno i file di configurazione rispettivamente per i siti *www.corp.local* e *intra.corp.local.*

Note: i nomi dei file devono terminare con "*.conf*"

Editare entrambi file creati specificando opportunamente i valori per le direttive:

```
ServerName[15] www.example.com
DocumentRoot /var/www/html
```

Inoltre per il sito *intra.corp.local* disabilitare l'*HTTP persistente* aggiungendo la direttiva:

```
KeepAlive Off
```

Ora occorre "abilitare" i siti configurati affinché siano visibili on-line, eseguire per entrambi i siti:

```
a2ensite nome_file_configurazione_sito
```

Infine, ricaricare la configurazione del server web con:

```
systemctl reload apache2
```

1. Da H1 (o R1) collegarsi con il browser alle homepage dei due siti e catturare il traffico generato.

Elencare le differenze più salienti tra le due catture.

Varia il contenuto del campo *Host* nell'header HTML nel pacchetto di risposta inviato dal server.

Il browser effettua sempre una richiesta di connessione persistente con l'header:

Connection: keep-alive

Nella risposta di *intra.corp.local* sono impostate le opzioni per HTTP non persistente con l'header:

Connection: close

Nel collegamento al sito *intra.corp.local* per ogni oggetto della homepage che ha un riferimento esterno viene creata una nuova sessione TCP e di conseguenza si ha un maggiore traffico di rete. Nel collegamento al sito *www.corp.local* invece una sessione TCP può venire utilizzata per effettuare GET multiple.

2. Scaricare la homepage di entrambi i siti tramite *telnet* e osservare se le due catture sono differenti rispetto al punto precedente. Specificare i comandi di protocollo necessari.

[14]Documentazione di rif. Apache2 https://httpd.apache.org/docs/2.4/vhosts/
[15]Documentazione di rif. Apache2 https://httpd.apache.org/docs/2.4/vhosts/examples.html

```
root@h1:~# telnet www.corp.local 80
Trying 192.168.0.129...
Connected to s1.corp.local.
Escape character is '^]'.
GET / HTTP/1.1
host:www.corp.local
```

Note:
i metodi http sono case sensitive – https://tools.ietf.org/html/rfc2616#section-5.1;
dopo il comando di header *"host:www.corp.local"* premere *Invio* **2 volte**

Le due catture sono sostanzialmente uguali, viene aperta una sola connessione TCP e richiesta la homepage con una GET. In entrambi i casi non si hanno connessioni multiple, ma in precedenza il browser analizzando il contenuto della pagina procede con nuove GET per gli eventuali riferimenti esterni presenti.

Seconda parte

In base alla configurazione del DNS fatta nell'esercitazione 2 il nodo S1 è indicato essere il server mail per il dominio *corp.local:* come server SMTP si utilizza *Postfix*[16], mentre per PO3/IMAP è usato *Dovecot*[17].
Nella configurazione utilizzata il sistema di posta fa uso degli utenti Linux presenti sul nodo S1 e non permette l'invio di messaggi ad utenti esterni.

Avviare Wireshark su R1 per osservare il traffico generato. Da H1 collegarsi con *telnet* al server mail, tramite protocollo SMTP inviare da *user1* una mail a *user2* e in seguito leggere sempre con *telnet* il messaggio utilizzando il protocollo POP3.

1. Indicare la porta utilizzata per l'ascolto del server SMTP e l'insieme minimo dei comandi che deve essere inviato al server per ottenere l'invio di un messaggio.

Invio SMTP.

```
root@h1:~# telnet mail.corp.local 25
Trying 192.168.0.129...
Connected to mail.corp.local.
Escape character is '^]'.
220 mail.corp.local - Reti di Calcolatori - ESMTP Postfix
HELO h1.corp.local
250 mail.corp.local
MAIL FROM:user1@corp.local
250 2.1.0 Ok
RCPT TO:user2@corp.local
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
SUBJECT:Reti di Calcolatori
ESERCITAZIONE 3
Protocolli di posta: SMTP, POP3
.
250 2.0.0 Ok: queued as 190B4255ED
quit
221 2.0.0 Bye
Connection closed by foreign host.
root@h1:~#
```

Nota: comandi e opzioni SMTP sono case-insensitive - https://tools.ietf.org/html/rfc5321#section-2.4

[16]http://www.postfix.org/
[17]https://www.dovecot.org/

2. Indicare la porta utilizzata per l'ascolto del server POP3 e l'insieme minimo dei comandi che deve essere inviato al server per la lettura di un messaggio.

Lettura POP3.

```
root@h1:~# telnet pop3.corp.local 110
Trying 192.168.0.129...
Connected to s1.corp.local.
Escape character is '^]'.
+OK Dovecot ready.
USER user2
+OK
PASS networks
+OK Logged in.
stat
+OK 1 362
list
+OK 1 messages:
1 362
.
retr 1
+OK 362 octets
Return-Path: <user1@corp.local>
X-Original-To: user2@corp.local
Delivered-To: user2@corp.local
Received: from h1.corp.local (h1.rd.corp.local [192.168.0.1])
      by mail.corp.local (Postfix) with SMTP id B802A255ED
      for <user2@corp.local>; Mon, 17 Jul 2017 10:33:46 +0200 (CEST)
SUBJECT:Reti di Calcolatori
ESERCITAZIONE 3

Protocolli di posta: SMTP, POP3
.
dele 1
+OK Marked to be deleted.
quit
+OK Logging out, messages deleted.
Connection closed by foreign host.
root@h1:~#
```

Nota: comandi e opzioni POP3 sono case-insensitive - https://tools.ietf.org/html/rfc1939#section-3

3. Osservando il traffico generato, quale fondamentale problema rende altamente insicuro il sistema in esame?

Si nota chiaramente che la password dell'utente *user2* transita in chiaro sulla rete!

4. Provare ad inviare un messaggio di posta
   - ad un utente esterno non appartenente al dominio *corp.local*;
   - ad un utente inesistente del dominio *corp.local*.

Quali codici di risposta vengono inviati dal server?

Mail a utente esterno:

454 4.7.1 <test@google.it>: Relay access denied

Mail a utente inesistente:

550 5.1.1 <user3@corp.local>: Recipient address rejected: User unknown in local recipient table